2020_군산대 교육 4주차
OpenCV

WeGo

03

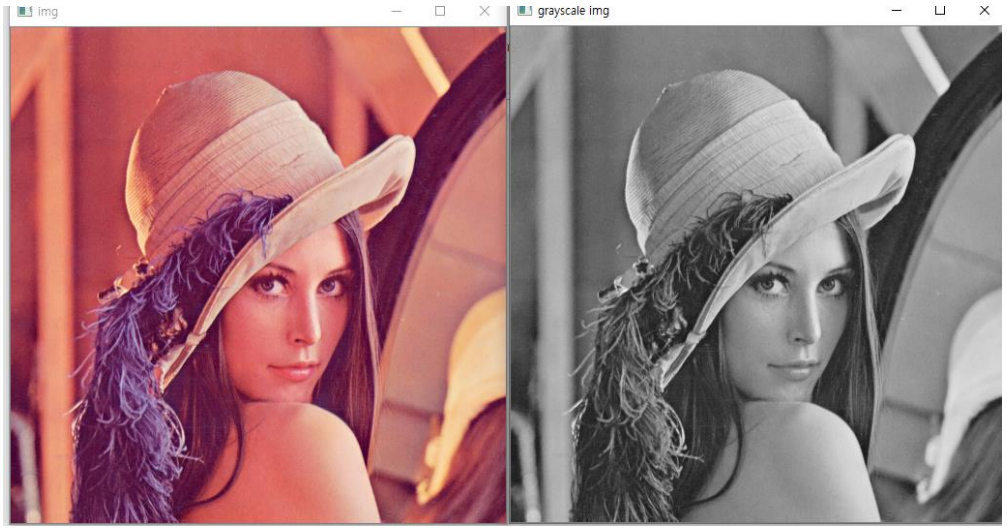
OpenCV 사용하기 II

03 그레이스케일

- **학습 목표: 이미지의 색상을 흑백으로 변경합니다.**
- **메인 코드:**

```
In [1]: import cv2

        img = cv2.imread("Image/Lenna.png", cv2.IMREAD_COLOR)
        grayscale_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

        cv2.imshow("img", img)
        cv2.imshow("grayscale img", grayscale_img)
        cv2.waitKey(0)
        cv2.destroyAllWindows()
```

- **결과:**



03 그레이스케일

- **코드 설명:**

1. cv2.cvtColor (이미지, 색상 변환 코드)

   이미지의 색상을 변경할 수 있습니다.

   색상 변환 코드는 원본 색상 코드 2 결과 색상 코드로 이루어집니다.

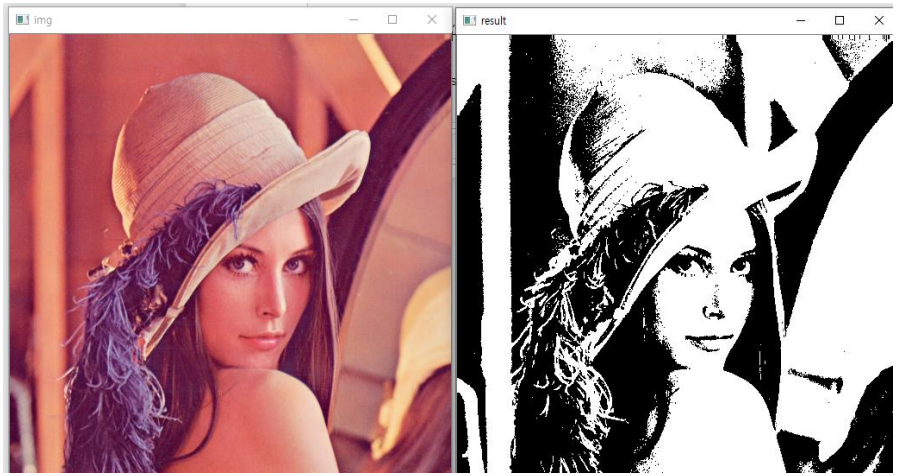
   ex) BGR2GRAY = BRG (RGB 역순) 2 GRAY (단일 채널 색상)

03 이진화

- **학습 목표: 이미지의 색상을 임계값을 기준으로 흰색 또는 흑생으로 변경합니다.**
- **메인 코드:**

```
In [18]: import cv2

img = cv2.imread("Image/Lenna.png", cv2.IMREAD_COLOR)

gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
ret, threshold = cv2.threshold(gray, 100, 255, cv2.THRESH_BINARY)

cv2.imshow("img", img)
cv2.imshow("result", threshold)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- **결과:**



03 이진화

- **코드 설명:**

1. cv2.threshold(그레이스케일 이미지, 임계값, 최대값, 임계값의 종류)

   그레이스케일 이미지: 그레이스케일된 이미지를 입력

   임계값: 이미지를 흑백으로 나누는 기준값

   최대값: 이미지의 최대값

   임계값의 종류: 임계값을 기준으로 이진화하는 방법

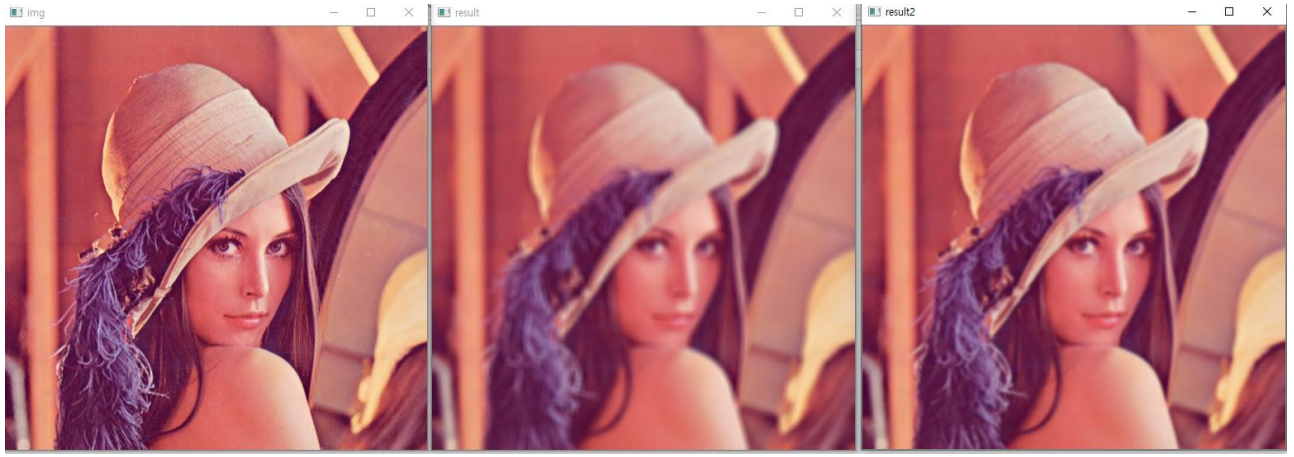
cv2.THRESH_BINARY	픽셀값이 지정한 임계값보다 크면 최대값, 작으면 0
cv2.THRESH_BINARY_INV	픽셀값이 지정한 임계값보다 크면 0, 작으면 최대값

03 Blur

- **학습 목표: 이미지에 흐림 효과를 넣어 번지게 합니다.**
- **메인 코드:**

```
In [*]: import cv2

img = cv2.imread("Image/Lenna.png", cv2.IMREAD_COLOR)
blur = cv2.blur(img, (9, 9), anchor=(-1, -1), borderType=cv2.BORDER_DEFAULT)
gaus = cv2.GaussianBlur(img, (9, 9), -1)

cv2.imshow("img", img)
cv2.imshow("result", blur)
cv2.imshow("result2", gaus)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- **결과:**



03 Blur

- **코드 설명:**

1. cv2. blur(원본 이미지, 커널 xy크기, 커널 중심점, 픽셀 외삽법)

   커널 xy크기: 흐림 효과를 적용할 크기. 크기가 클수록 더 흐려짐

   커널 중심점: 커널에서 중심점을 뜻하며, (-1,-1)은 기본 중심점으로 할당됨

   픽셀 외삽법: 이미지를 블러 처리할 때, 영역 밖의 픽셀은 추정해서 값을 할당

cv2.BORDER_DEFAULT	gfedcb | abcdefgh | gfedcba
cv2.BORDER_CONSTANT	iiiiii | abcdefgh | iiiiiii

2. cv2.GaussianBlur (원본이미지, 커널 xy크기, 표준 편차)

   표준 편차: 가우시안 커널의 표준 편차를 뜻함. 0으로 설정하면 커널 크기를 고려해 자동 설정

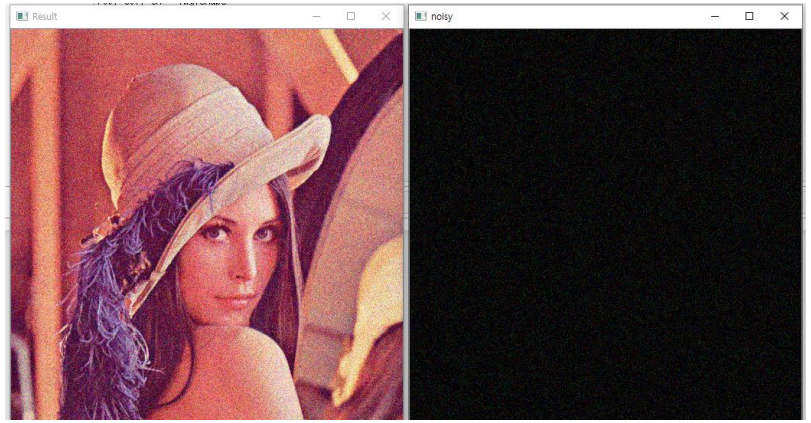
03 노이즈 생성

- **학습 목표: 원활한 블러 처리를 위해 노이즈를 생성하기.**
- **메인 코드:**

In [*]:
```
import cv2
import numpy as np

img = cv2.imread("Image/Lenna.png").astype(np.float32)/ 255.0

row, col, ch = img.shape

mean = 0
sigma = 0.1

gauss = np.random.normal(mean, sigma, (row, col, ch))
gauss = gauss.reshape(row, col, ch)
noisy = img + gauss

cv2.imshow('original', img)
cv2.imshow('Result', noisy)
cv2.imshow('noisy', gauss)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

03 노이즈 생성

- **결과:**



- **코드 설명:**

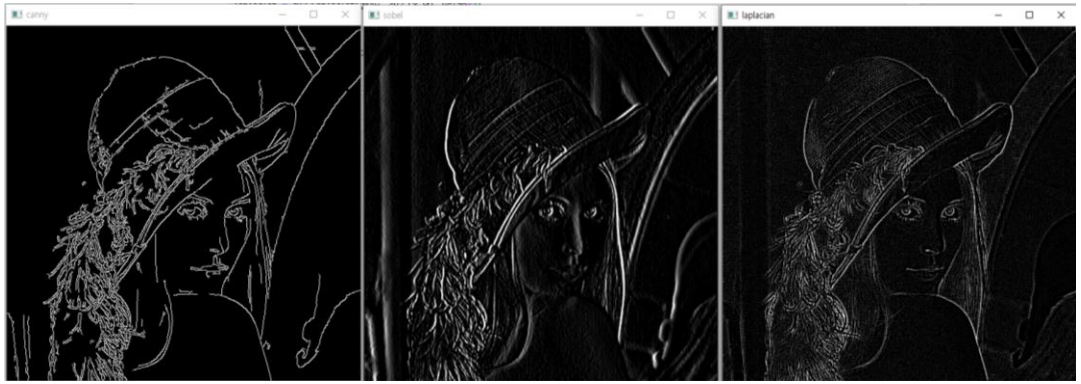
1. astype(np.float32)/255.0 : 색상값을 0~255에서 0~1사이의 부동소수형으로 변경
2. np.random.normal(mean, sigma, (row, rol, ch)) : 정규분포를 만들어주는 함수로써 인자 값은 순서대로 평균, 표준편차, 크기입니다.(쉽게 말해 무작위 값 생성)
3. reshape(row, col, ch) : 생성된 무작위 값을 이미지와 같은 행렬 형태로 변환 시켜주는 함수
4. noisy = _image + gauss : 2개의 이미지를 합쳐주는 코드입니다.

03 Edge

- **학습 목표: 이미지의 가장자리(Edge)를 검출하기 위해 사용합니다.**
- **메인 코드:**

In [*]:
```
import cv2

img = cv2.imread("Image/Lenna.png", cv2.IMREAD_COLOR)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

canny = cv2.Canny(img, 100, 255)
sobel = cv2.Sobel(gray, cv2.CV_8U, 1, 0, 3)
laplacian = cv2.Laplacian(gray, cv2.CV_8U, ksize=3)

cv2.imshow("canny", canny)
cv2.imshow("sobel", sobel)
cv2.imshow("laplacian", laplacian)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- **결과:**



- **코드 설명:**

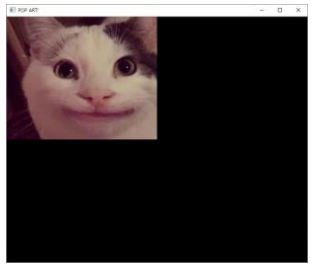
1. cv2.Canny(원본 이미지, 임계값1, 임계값2, 커널 크기, L2그라디언트)를 이용하여 가장자리 검출을 적용합니다.
   임계값1은 임계값1 이하에 포함된 가장자리는 가장자리에서 제외합니다.
   임계값2는 임계값2 이상에 포함된 가장자리는 가장자리로 간주합니다.
   커널 크기는 Sobel 마스크의 Aperture Size를 의미합니다. 포함하지 않을 경우, 자동으로 할당
   L2그라디언트는 L2방식의 사용 유/무를 설정합니다. 사용하지 않을 경우, 자동적으로 L1그라디언트 방식을 사용합니다.

03 np.zeros

- **학습 목표: 검은색 캔버스 화면 생성하기**
- **메인 코드:**

```python
import cv2
import numpy as np

img = cv2.imread("image/cat.jpg")
height, width, channel = img.shape

face_image = img[int(height/3) : int(height * 2 / 3), 0:int(width/2)]

height, width, channel = face_image.shape

canvas = np.zeros((height*2, width*2, channel), np.uint8)
canvas[0:height, 0:width] = face_image

cv2.imshow("POP ART!", canvas)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

- **결과:**



03 문제 2. GrayScale, Canny, Blur 활용하기

- **문제:**

  순서는 상관없이 원본, GrayScale, Canny, Blur 이 4가지 얼굴을 팝아트로 그려주세요.

  필수 hint. 흑백 -> RGB 이미지 변환 = cvtColor(이미지, cv2.COLOR_GRAY2BGR)

- **결과:**